Proposition for propagated occupation grids for non-rigid moving objects tracking

Benjamin Lefaudeux, Gwennaël Gate, Fawzi Nashashibi

Proposition for propagated occupation grids for non-rigid moving objects tracking

Benjamin Lefaudeux
IMARA - INRIA Rocquencourt
benjamin.lefaudeux@inria.fr

Gwennael Gate
Aldebaran Robotics
Email: ggate@aldebaran-robotics.com

Fawzi Nashashibi
IMARA - INRIA Rocquencourt
CAOR - Mines Paristech
Email: fawzi.nashashibi@inria.fr

***Abstract*—Autonomous navigation among humans is, however simple it might seems, a difficult subject which draws a lot a attention in our days of increasingly autonomous systems. From a typical scene from a human environment, diverse shapes, behaviours, speeds or colours can be gathered by a lot of sensors ; and a generic mean to perceive space and dynamics is all the more needed, if not easy. We propose an incremental evolution over the well-known occupancy grid paradigm, introducing grid cell propagation over time and a limited neighbourhood, handled by probabilistic calculus. Our algorithm runs in real-time from a GPU implementation, and considers completely generically space-cells propagation, without any a priori requirements. It produces a set of belief maps of our environment, handling occupancy, but also items dynamics, relative rigidity links, and an initial object classification. Observations from free-space sensors are thus turned into information needed for autonomous navigation.**

I. Occupancy grids - extended

A. General considerations

1) Perception of non-structured moving objects: Amid a vast set of localisation and perception algorithms currently developed, among which various SLAM variants have taken the greater share for the last years, some challenges are still not addressed in the perception field. Detection and tracking of moving objects, without pre-requisites on rigidity, structured environment or immobility remains a tough question, and is not yet addressed by many approaches. Different SLAM-based approaches have been used for the last years, some of them being very successful in building a map of the environment from a non-associative sensor, while dealing with the identification of some moving objects amid the map. FastSLAM, using particle filtering and initially presented by Thrun et. al ([1]) or other iterative optimisation techniques (ICP, [2] [3]) are very powerful algorithms which can deal with some level of movement amid tracked points, as long as fixed points of the map are enough to estimate robot pose and environment mapping. This proved effective, for example by Wang et. al ([4]), which showed that moving points could be separated from a still structure with a good accuracy, and tracked accordingly. Overall scene requisites are however still strong in this setup, and would probably not be enough for reliable moving object perception and tracking in a highly unstructured environment, without a majority of solid points to observe, which can easily happen outdoor in densely populated areas. Another possibility is exploitation of detection before tracking, using information from visual or speed measures sensor, in order to remove moving objects from the traditional SLAM calculus. This has notably been demonstrated by Agrawal ([5]), which proved effective to detect individuals while still needing fixed points as a reference. Contrary to SLAM, our aim is thus not primarily to localise oneself in the environment along with building a map of its static features, but instead to be able to reliably detect and monitor position and dynamics of surrounding moving objects, possibly outdoor and without much visible solid infrastructure. Those observations will drive the following developments, and while the proposed algorithm is nowhere near best SLAMs in terms of accuracy, we believe it could be useful in typical human environments.

2) Interests and limits of the simple grid-based approach: Occupancy grids are very common in robotics, since their first introduction by Moravec and Elfes ([6]), initially related to sonar based mapping. Principles are simple and effective, relying on the information storage and information source spatial localisation similarities, thus allowing to keep any spatial relations between cells at a minimal cost. This is still used nowadays, although not being any more the only processing step of the algorithm, and can still be viewed as a very-capable mean of storing spatially related information or sharing information between several subjects. It is also specially well-suited to deal with promising new massively-parallel computing capabilities. Occupancy grids were indeed present in most of the latest SLAM propositions (based on filtering or optimisation), and in many of broadly speaking perception systems (Badino et al [7] showed for example a free-space perception system based on vision and occupancy grids in 2007, as did more recently by Yguel et al. [8]).

In those examples, grids are however only used for information representation and storage, most of the processing being due to external algorithms. Sensor filtering over time is indeed rarely considered within the grid formalism (grid update rule often relies on mere accumulation). Links between cells are also rarely considered while updating grids, spatial and temporal independence between measures being at the heart of the initial simplicity of the method. Obviously a big step from accurate world description, this is basically of little consequences provided the aim is, as it was in the original Moravec article [6], the cartography of still environment. In case we chose the occupancy grid formalism to track moving objects, this is however a major limitation, as one could expect the displacement of physically related points to be correlated.

3) Optimal update to occupancy grids: Considering a hypothetical occupancy grid propagation, one could wonder why an optimal Bayesian propagation would not be possible. The probability of any possible cell displacement could be computed, thus computing the most-probable map prediction from a given set of measure, allowing both temporal filtering and a link between new measures and known information. Trouble is, computing the formal probability (provided all the needed information is known) is extremely demanding, if one is to consider every possible move. A complete probability calculus, considering a set of S possible state in any cell of a $N*M$ grid, would imply the consideration of each and every possible map, which is S^{N*M}. Taking the smallest possible state sampling (2 states), and a ridiculously small map of 10*10 cells, and the amount of maps to compute to get to the full solution is already beyond any reasonable range (2^{100}).
The key to this "absurd" complexity when keeping every probability on track, which could seem somewhat odd from a human perception (we don't usually need much time to get a good grasp on a typical scene involving moving objects), surely relies in a lot of useless probabilities being taken into account, even possibly beyond the causality principle. Anything at a given place have little to no chance of influencing the very next future of a very remote location, and this is the idea behind dealing with neighbourhood-restricted probabilities.
Another possible approach, demonstrated by Coue et. al in [9], would be the use of a Bayesian network coupled with a motion model for dynamic objects mapping. In this case, no actual association needs to be done, this being handled by the Bayesian framework computing transition probabilities between states. The main difference with our approach is that they do not attempt to compute the most probable next place for a given occupancy, but rely on a given motion model (constant velocity in this example). This is very fine in most cases, and definitely is an improvement over static occupancy maps, but we believe that this can lead to the wrong prediction in some cases, among which heavy occlusion or colliding courses. We however certainly share a lot of the abstraction presented in this article, although a few more notions are present in our algorithm.

B. Tackling inter-dependence within computing boundaries

We propose the use of propagated occupancy grids, able to deal with some of the interactions between cells, in a common prediction/measure Bayesian cycle. Our algorithm aims at taking into account both temporal and spatial relations between measures, while keeping computing costs low enough to conceive a real-time use and concurrent use to other more specialised algorithms.

Firstly, we introduce the probability for every cell to move to its neighbourhood, given previous knowledge of the scene (occupation, speed, classification) and specific heuristics (separate cells cannot converge, nor can cells from the same object diverge). Secondly, we compare this prediction to a new measure, and compute the most probable estimate given prediction and latest measure. Thirdly, we update associated knowledge used in the prediction step, namely occupancy of every cell of the grid, speed, relation between cells (in a neighbourhood) or object classification used for different sensor models. Those principles were presented by Gate in [10], initially on a standard CPU implementation, and showed very promising results despite a high computing cost making it prohibitive for any real-time application.

1) Initial definitions : : The probability mass functions (*pmf*) modelled in the following algorithm concern a set of notions that we'd like to define :
- Mapping (occupancy) probability $M_k(x_i) \in [0,1]$ of the cell x_i in the spatial environment E at the time *k*, provided the measures $Z_{0:k} = \{z_0, ..z_k\}$:

$$P(M_k(x_i) = 1 | Z_{0:k}) \quad \forall x_i \in E \tag{1}$$

- Vehicle localisation (including position and speed in $E \times V$), at the iteration *k*. This is not yet addressed by the algorithm, and in the examples below every speed and position is relative to the vehicle.

$$P(L_k = l_j | Z_{0:k}) \quad \forall l_j \in (E \times V) \tag{2}$$

- Association, *ie* the probability for a given cell from the iteration *k-1* to be associated with another given cell at the iteration *k*. In our case, only associations coming from a restricted neighbourhood are taken into account, which cuts the number of evaluated map candidates from an exponential dependence on the number of cells to a more reasonable dependence on neighbourhood scale.

$$P(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_i) = 1, Z_{0:k}) \quad \forall (x_i, x_j) \in E^2 \tag{3}$$

- Velocity probability of a cell, given its occupancy and previous measures :

$$P(V_k(x_i) = v | M_k(x_i) = 1, Z_{0:k}) \quad \forall (x_i, v) \in E \times V \tag{4}$$

- Detection probability, to handle the probability that two given cells x_i and x_j are part of the same object. The neighbourhood constraint limiting interactions to a finite part of the map is used once more, to limit the intricateness and heavy computing cost. Affiliation to a given object can however be "propagated" further than one cell's neighbourhood, although our span is limited and this could prove to be a problem. Detection probability, $D_k(x_i, x_j) \in [0,1]$ is 1 if x_i x_j are from the same object.

$$P(D_k(x_i,x_j)=1|M_k(x_i)=1,M_k(x_j)=1,Z_{0:k}) \quad \forall(x_i,x_j)\in E^2 \tag{5}$$

Several criteria could be used to determine this probability from measures, among which a constant relative configuration, fit with a given shape, or a set of distinct characteristics (speed, shape, colour,..). In this first implementation, a simple geometrical criteria is used : $D_k(x_i,x_j)=1$ if the distance between x_i and x_j is conserved over iterations, with a Gaussian decrease elsewhere (detailed below).

- Classification designs the probability of this set of cells to be part of a given class of objects (car, pedestrian, still object,..). We attempt to model this by matching extended characteristics of a set of cells (beyond geometrical characteristics for example) to a model. This conditions the update rule of association calculus, envisaged future positions of a cell being for example adapted from its class motion model. Classification probability are simultaneously kept from a different set of classes, a cell being capable of a partial fit with different classes. This ensures a more robust classification, initially prone to errors.

$$P(C_k(x_i)=c_j|M_k(x_i)=1,Z_{0:k}) \quad \forall(C\times E) \tag{6}$$

2) Update rules - proposed algorithm : : Having set these definitions, the proposed update rules, implemented as such in the parallel execution we present as a last-part example, are as follows :

- Associations are updated in a several pass mechanism, making an extra initial assumption of independence between cells behaviour, which we attempt to correct in a second part. This behaviour was already present in the [10] proposition, and is a key to the possible use of massively parallel computing. Approximations are obviously primordial in our attempt to make the calculus feasible in real-time, but we believe most of the interactions between cells are still modelled with this proposition.

$$\begin{aligned}P_{local}(X^{next}_{k-1}(x_j)|M_{k-1}(x_j)=1,Z_{0:k}) = \\ \eta\cdot\underbrace{P_{local}(z_k|X^{next}_{k-1}(x_j),M_{k-1}(x_j)=1,Z_{0:k-1})}_{Correction} \\ \cdot\underbrace{P_{local}(X^{next}_{k-1}(x_j)|M_{k-1}(x_j)=1,Z_{0:k-1})}_{Prediction}\end{aligned} \tag{7}$$

η is here a normalisation constraint, to ensure that possible moves sum up to one for any given cell.

First we then compute the local associations prediction, which is to say that macroscopic interactions are not yet taken into account : cell previous speed and class are used to predict the asserted new positions. This could be seen, similarly to SLAM particle filters ([1]), as a new set of particles generated for every cell of the grid iteratively, representing this cell's occupancy possible next moves, depending on previous knowledge and motion model. In our case, initial predictions are weighted by a Gaussian, whose standard deviation is function of the identified class of the object (thus representing the possible uncertainty in an object next move, this being different for a pedestrian or an identified bus). The centre of the Gaussian weight is relative to the object previous speed estimation.

$$\begin{aligned}P_{local}(X^{next}_{k-1}(x_i)=x_j|M_{k-1}(x_j)=1,Z_{0:k-1}) = \\ \Psi(x_j,x_i,V_{k-1}(x_j),C_{k-1}(x_j))\end{aligned} \tag{8}$$

which could be rewritten as, with g the standard Gaussian expression and σ its standard deviation dependent on the identified class :

$$\begin{aligned}\Psi(x_j,x_i,V_{k-1}(x_j),C_{k-1}(x_j)) = \\ g(x_j+\mathbf{V}_{k-1}(x_j)\cdot dT - x_i, \sigma_{C_{k-1}(x_j)})\end{aligned} \tag{9}$$

Last measure is then taken into account to produce an estimated local association, still without macroscopic constraints to alter these predicted associations. Predictions are weighted according to the sensor model occupancy new measure, while keeping the normalized sum of all possible displacements :

$$\begin{aligned}P_{local,weighted}(X^{next}_{k-1}(x_i)=x_j|M_{k-1}(x_j)=1,Z_{0:k}) = \\ \gamma P_{local}(X^{next}_{k-1}(x_i)=x_j|M_{k-1}(x_j)=1,Z_{0:k-1}) \\ \cdot P(M_k(x_j)=1|Z_k)\end{aligned} \tag{10}$$

with γ such as $\sum_{a_k\in \mathrm{A}} P_{local,weighted}(a_k)=1$ (A being the set of investigated associations).

Associations initially local estimation are then altered according to additional constraints : unlikely moves are penalised according to different heuristics (different cells cannot converge to the same place, cells from the same rigid object cannot diverge). Rigidity and non inter-penetration constraints are modelled by the potential function $\Phi_{association}$, which is currently based on two Gaussian-window weight functions.

$$\begin{aligned}P(X^{next}_{k-1}(x_j)=\hat{x}|M_{k-1}(x_j)=1,Z_{0:k-1}) \\ \simeq \sum_{a_k\in\mathrm{A}} \\ \left\{\prod_{1\le j\le N} P_{local}(X^{n}_{k-1}ext(x_j)|M_{k-1}(x_i)=0,Z_{0:k})\right\} \\ \cdot\Phi_{association}(a_k,E,Z_{0:k})\end{aligned} \tag{11}$$

were A is again the set of all possible associations. The process here described can be differently factorised, but was split into several summations in an attempt to increase its readability.

- Mapping is computed taking into account the two cases : in the cell is a newly observed presence, or the displacement (possibly null) of a previously seen cell. In the first case, the sensor model is the only input taken into account, in the second case contributions from all the possible associations are summed up to compute the predicted occupancy. Cell interactions have in this case already been taken into account in the association computation. The two cases are dissociated by a random variable S_k, which can take two values : 0 if the cell has never been seen, 1 if the cell has already been seen. Its probability is computed with association computation results : to sum up, if the considered cell corresponds to a local association maximum, $P(S_k(x_i) = 1|Z_{0:k})$ takes the value $\gamma \in [0, 1]$, else it takes the value $1 - \gamma$. The value of γ is chosen depending on the "renewal" rate of the map, that is to say "how often do we think a new object can appear from nowhere" ? A value around 0.5 has proven to work well in practice.

$$
\begin{aligned}
&P(M_k(x_i)|Z_{0:k}) = \\
&\quad P_{seen}(M_k(x_i)|S_k x_i = 1, Z_{0:k}) \cdot P(S_k(x_i) = 1|Z_{0:k}) \\
&\qquad + P_{unseen}(M_k(x_i)|S_k x_i = 0, Z_{0:k}) \\
&\qquad\quad \cdot P(S_k(x_i) = 0|Z_{0:k})
\end{aligned} \tag{12}
$$

with the corresponding calculus (A being the cell neighbourhood) :

$$
\begin{aligned}
&P_{seen}(M_k(x_i)|Z_{0:k}) = \\
&\sum_{j \in A} \{P(X^{next}_{k-1}(x_j) = x_i|M_{k-1}(x_j) = 1, Z_{0:k-1}) \\
&\quad \cdot P(M_{k-1}(x_j) = 1|Z_{0:k-1})\}
\end{aligned} \tag{13}
$$

P_{unseen} is in this case typically related to the sensor occupancy model.

- Velocities are computed taking into account the same two possibilities, depending if the observed cell is considered a new one, or the association of an already-observed cell to a new position :
- considering the velocity of already-observed cells, velocity is simply computed from the associations, summing up speed values steming from all the possible contributors.
- considering appearing cells, the probability distribution of velocities had been proposed by Gate in [10] as follows, and kept in this proposition :

$$P(V_k(x_i)|S_k(x_i) = 0, Z_{0:k}) = \frac{1}{card(V)} \tag{14}$$

Merging of the two possibilities is done similarly to *eq.* 12. In current implementation, we only retain the most probable *a posteriori* value, and a confidence value, instead of retaining the whole probability distribution.

- Detection update needs a broader view from previous information, in order to be able to detect structures and links between updated cells. In this paper we propose a simple (and fast) mechanism to handle this detection, but we expect to adopt a maybe more "large scale" approach in the future. The mechanism proposed in this initial algorithm consists in measuring the auto-correlation between consecutive cells associations after filtering, in the vicinity of their neighbourhood. We then measure the compatibility of their predicted moves with the "rigid body" hypothesis. With the approach used in *eq.* 12, we dissociate in this calculus cells which are believed to have been seen before from "new" ones, for which no rigidity information can be guessed.

$$
\begin{aligned}
&P(D_k(x_i, x_j) = 1|M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k}) = \\
&\sum_{a_k, a'_k \in A_k \times A'_k} \\
&\{P(X^{next}(x_i) = a_k|Z_{0:k}) \cdot P(X^{next}(x_j) = a'_k|Z_{0:k})\}
\end{aligned} \tag{15}
$$

with a_k, a'_k being in fact the same associations in each cell respective referential (we go through every possible association for the (x_i, x_j) cells. Although not very illustrative, *eq.* 15 calculus is fast to compute, but relationships further than a cell neighbourhood are not taken into account (typically a few meters radius). This could prove insufficient for the tracking of big objects, and other methods could be investigated in the future. A simple k-means clustering could for example be used outside of the prediction/update cycle to emphasize object detection for an external navigation task.

- Classification updates can similarly be done using every gathered information (mapping, velocity, rigidity links,..) correspondence to a given sensor model, which would on the other hand improve prediction steps of the algorithm. This is not yet present in our implemented algorithm, and presented results can thus be seen as perfectible. There is however no theoretical constraints on this calculus, which should be in place in our implementation algorithm in a short time for several classes of objects (pedestrian, cars,..). The performance impact is to be investigated, but should not theoretically prevent the algorithm to run in real-time, every added class acting in this organisation as another "layer" of probabilities to be computed. Worst impact could thus be a linear cost in terms of the number of identified classes, which should not be increased inconsiderately.

II. Practical implementation and results

A. Performance considerations

As stated in *eq.* I-A3, complexity constraints on calculus are not to be neglected in occupancy grid update rules, many thinkable algorithms being simply not realistic in terms of computing needs. The complexity of the mechanism we propose can be summarised as follows :
- for the sake of simplicity, we state a $N * N$ grid, every cell being able to move in a $M * M$ neighbourhood.
- considering the propagation of one cell, every possible move of every of its neighbours ($O(M^4)$) are to be investigated for each individual envisaged propagation ($O(M^2)$), which translates in $O(M^6)$ complexity.

- cells updates being independent except for conformation rules already taken into account in the previous step, overall cell considerations is finally of $O(N^2 * M^6)$ complexity for the full update step.

Heavy approximations are of course still present when compared to an full propagation calculus, namely that each and every cell moves are initially computed individually, although being later filtered to take some interactions into account. Computation is thus still not intricate, which keeps the complexity "low", although $O(N^2 * M^6)$ remains a heavy burden for any realistic dimensions. Several points can be emphasised from this simple complexity calculus : Firstly, although the computation remains slow by all means, the relative independence of most calculus make it a plausible candidate for a parallel implementation, which is nowadays common and lifts some of the computing time constraints. Secondly, digging into specific complexity aspects, our algorithm is also linear in complexity as regards the number of cells for a given neighbourhood (which translates into a squared dependence as regards the size of the grid, naturally). This seems quite a burden, but one must remember the usual exponential complexity of algorithms meaning to explore any possible point move onto a map. This relative lightness in the grid size complexity is a key benefit for sharing applications : extending the size of the grid translates into linear increase in complexity instead of an exponential increase in case of a propagation calculus on the whole map. Extending the domain of tracked speed is however more complicated : tracked speed depends on the span of possible moves taken into account at every iteration, literally $V_{max} = \frac{M/2*\Delta}{\tau_{iteration}}$ with Δ the spatial extension relative to one grid cell. Considering a given iteration maximum computing time (limited for example by the Lidar frequency for real-time operations), the maximum tracked speed is rapidly capped by the computing power at disposal relatively to the maximum computing time. We'll see with our preliminary results that this translates to very acceptable maximum speeds for our initial implementation on current hardware.

B. Some results

As usual when dealing with grid-based algorithms, sensor occupancy models are a key factor in our proposition. A standard Lidar occupancy model is used in this initial implementation, computed on GPU. Occupancy of areas in the shadow of laser impacts are chosen to 0.5 out of 1, neutral in our occupancy ratio. Laser impacts are otherwise set to an occupancy probability of 1, while empty spaces between the vehicle and impacts are set to 0, as it is common using Lidars. Real data gathered in an urban environment are used. Logging and replay framework is RTMaps software from Intempora.
Exhaustiveness and generic nature of the algorithm is important : every mapped cell of the environment is considered equal, and no *a priori* is ever made on geometrical bounding, preferred positions, structures, stationary or moving parts. Although still quite demanding on computing resources, the algorithm works in real-time on current state-of-the-art hardware. The first scene computed below needs 120ms to compute on a GF100 GPU from nVidia counting 448 cores, which translates without additional work to below 100 ms on higher-end offerings currently available. As regards raw performance, this algorithm is also an initial draw, and pure implementation could certainly be greatly improved as it is often the case with hardware-sensitive programming.

1) Algorithm memory: In this example, we emphasise a temporary occlusion situation, where a pedestrian shadow hides another previously seen pedestrian (figure *eq.* 1). Only the Lidar sensor is used in this case, camera captures being presented for illustrative purpose, along with bounding boxes. On the *pmf* representations (figures *eq.* 2 and *eq.* 3), the point of view is from above, in a common "bird-view" perspective. All the boxes are drawn for illustrative purpose, we don't present here the output of a detection algorithm. The resolution of grid mapping is 15cm, speeds up to 4.5m/s being theoretically tracked. This last value can be improved without any computing cost by simultaneously degrading the spatial resolution and increasing the range of the measures, which could be a dynamic trade-off depending on the vehicle speed.

(a) (b)

Fig. 1. Successive camera views

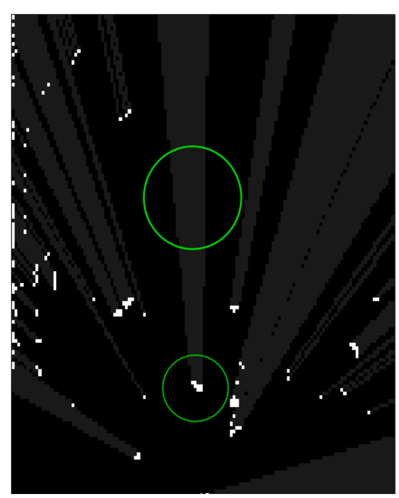

Fig. 2. Lidar output when occluded

The shadowed pedestrian is still clearly visible on the

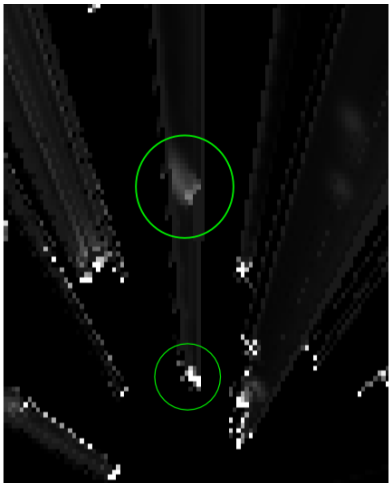

Fig. 3. Occupancy evaluation from our algorithm

occupancy map, although the Lidar cannot get through the first occluding person. Its position spreads over time, which shows that our knowledge decreases with the age of the data. Extensive propagation of cells once perceived occupied naturally leads to this result : occupied cells have not disappeared, although not being visible on the sensor. We thus stress the importance of sensor filtering, taking into account possible spatial and temporal correlation. In this case, consequences of the pedestrian not being visible without filtering has no practical consequences, but this is not always the case and we believe that such filtering would be compulsory for autonomous vehicle navigation in an urban area.

2) Dynamics estimation: This example plans to emphasise dynamics estimation capabilities of the algorithm, along with segmentation of the scene. Although this is already possible via bounding boxes in the case of clearly separated persons, those matchings often miss when persons are too close to each other, or when groups change in size due to some people joining and leaving. The exhaustive approach that we carry on a per-cell basis provides an estimation of the probable speeds. All boxes presented on figures *eq.* 4 *eq.* 5 *eq.* 6 are here on illustrative purpose, and do not come from a detection algorithm.

Fig. 4. Camera capture

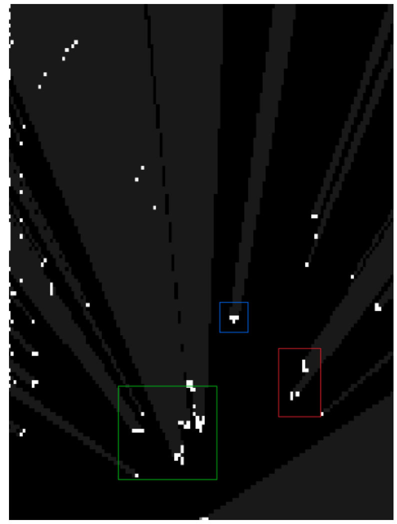

Fig. 5. Lidar sensor output

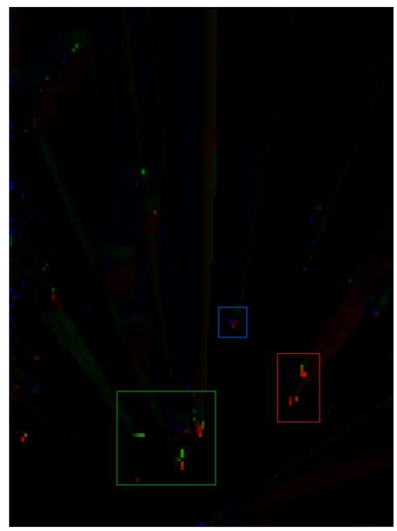

Fig. 6. Output of the algorithm - speed map

Figure *eq.* 6 shows the speed map maintained by the algorithm, speed orientation being coded by colour, while speed value (in the car referential) is represented by the brightness. Three boxes have been overlaid by hand on the figure, to emphasise specific cases. In green are the crossing pedestrians, which may have been difficult to track on sensor data alone (cf figure *eq.* 5), due to the pushchair and their proximity. In blue is the road sign, which is obviously standing still, but which shows the residual speed of the vehicle (end of the braking sequence) and noise on Lidar data. It is barely visible on the speed map, due to the very low residual speed of the car. Segmentation of the scene after the algorithm process between moving and still parts proves effective. In red is the car coming on the other way, which is also going slow due to the crossing pedestrians, and have very few Lidar impacts (laser beam was oriented upwards, maybe too much). Tracking on geometrical grounds on Lidar data alone may have been difficult in this case.

C. Future works

Alternative source of occupancy, velocity or classification are planned, mainly based on vision processing. Stereo-vision is for example a proven source of free-space measure, as shown by *Moravec* once again in [11], or more recently *Badino et al.* using dense disparity calculus and occupancy grids ([7] and articles following). Motion detection and evaluation have also be proven to be a valuable output from stereo-vision capture, *Argrawal et al.* demonstrating in [5] that platform ego-motion could also be removed from the initial optical flux in order to track moving objects. Initial theoretical work on this use would be from *Adiv et. al* ([12]), although this field has received a lot of attention in the past years, notably since dense stereo-vision processing is now possible in real-time. To finish with, the state of the art as regards vision-based SLAM (notably *Davison* [13] [14])) leads us to believe that laser-based sensors could possibly be replaced in the near future for most perception tasks, this being a strong incentive for us to develop visual inputs. Another source of possible improvements would be an evolution to handle collaborative perception, by means of merging grid-based beliefs in our algorithm.

III. Conclusion

Perception of moving objects, such as pedestrians, with minimal requirements on their size, moves or behaviour is a difficult task ; which will nevertheless be a key to enable autonomous navigation in urban environments, or even comprehensive assistance on current automotive devices. We proposed a novel technique based on an extension of the traditional occupancy grids registration, using probabilist propagation and extensive consideration of displacements and interactions over a restricted neighbourhood.

An evolution from an initial ambitious proposition, our work shows promising results while being capable of real-time execution, although still being a work-in-progress. An interesting evolution would be its extension towards collaborative perception, which should be easier than for some other approaches due to the grid-based principle being kept. Many challenges would still need to be resolved in order for separated vehicles to take part into one another perception of the environment, among which localisation or synchronisation inaccuracies, but we believe that autonomous transportation and road safety would benefit a lot from such developments.

References

[1] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, vol. 4, no. 3, pp. 380–407, 2004. [Online]. Available: \wrapletters{http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.6632\&rep=rep1\&type=pdf}

[2] C.-c. Wang and C. Thorpe, "A Hierarchical Object Based Representation for Simultaneous Localization and Mapping A Hierarchical Object Based Representation for Simultaneous Localization and Mapping," *Robotics*, 2004.

[3] B. Steux and O. El Hamzaoui, "CoreSLAM: a SLAM Algorithm in less than 200 lines of C code," *Submission ICARCV*, 2010. [Online]. Available: \wrapletters{http://www-roc.inria.fr/imara/dw/_media/users/oussamaelhamzaoui/coreslam.pdf}

[4] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007. [Online]. Available: \wrapletters{http://ijr.sagepub.com/cgi/doi/10.1177/0278364907081229}

[5] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Motion and Video Computing, 2005. WACV/MOTIONS'05 Volume 2. IEEE Workshop on*, vol. 2. IEEE, 2007, pp. 207–214. [Online]. Available: \wrapletters{http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4129607}

[6] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 116–121. [Online]. Available: \wrapletters{http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087316}

[7] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*. Rio de Janeiro: Citeseer, 2007, pp. 1–12. [Online]. Available: \wrapletters{http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.4161\&rep=rep1\&type=pdf}

[8] M. Yguel, O. Aycard, D. Raulo, and C. Laugier, *Grid based fusion of off-board cameras*. IEEE. [Online]. Available: \wrapletters{http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1689641}

[9] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006. [Online]. Available: \wrapletters{http://ijr.sagepub.com/cgi/doi/10.1177/0278364906061158}

[10] G. Gate and F. Nashashibi, "An approach for robust mapping, detection, tracking and classification in dynamic environments," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6. [Online]. Available: \wrapletters{http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5174739}

[11] H. Moravec, "Robot Spatial Perceptionby Stereoscopic Vision and 3D Evidence Grids," *Perception*, no. September, 1996. [Online]. Available: \wrapletters{http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.9045\&rep=rep1\&type=pdf}

[12] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 384–401, Jul. 1985. [Online]. Available: \wrapletters{http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767678}

[13] A. J. Davison and N. Kita, "3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain (PDF)," *Measurement*, vol. 00, no. C, 2001. [Online]. Available: \wrapletters{http://www.computer.org/portal/web/csdl/doi/10.1109/CVPR.2001.990501}

[14] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," London, pp. 0–7, 2003.